

Mathematics for Software Engineering

Authors: Richard Brooks & Eduard Fekete

Date: May, 2025

Version: 2.0

Contents

Chapter 1 Set Theory	1
1.1 What is a Set?	1
1.2 Important Sets: The Number Systems	2
1.3 Relationships Between Sets	4
1.4 Properties of Sets	6
1.5 Operations on Sets	7
1.6 Cartesian Products and Tuples	10
1.7 Proving Set Equalities	11
1.8 Computer Representation of Sets	13
Chapter 2 Combinatorics and Probability Theory	15
2.1 Sample Space and Events	15
2.2 Types of Events in Probability Theory	16
2.3 Counting Principles	18
2.4 Probability Basics	19
Bibliography	22
Appendix A Summation	23

Chapter 1 Set Theory

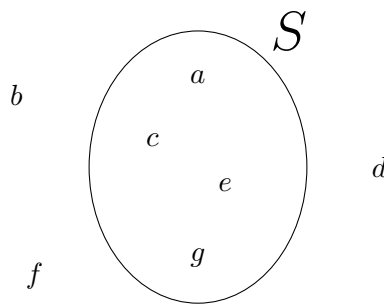
In software engineering, we are constantly working with collections of things: users in a system, records in a database, or nodes in a network. Set theory provides the formal mathematical language to describe and manipulate these collections with precision and clarity. It is the bedrock upon which many core computer science concepts—from database query languages like SQL to data structures and algorithmic logic—are built.

This chapter introduces the fundamental principles of set theory. We will begin with the simple, intuitive idea of a set and explore the formal notation used to define them. We will then cover the essential relationships between sets, such as subsets, and the core operations used to combine them, including unions, intersections, and complements. These operations directly correspond to the logical operators (OR, AND, NOT) that govern the flow of your code. Finally, we will explore ordered collections called tuples and introduce a simple method for proving set equalities.

1.1 What is a Set?

A set is an unordered collection of distinct objects. The objects within a set are called its **elements** or **members**. We can think of a set as a simple container where items are grouped together, and the order in which we list them does not matter. For example, the set of primary colors can be written as {red, yellow, blue} or equally as {blue, red, yellow}.

Sets are a cornerstone of modern mathematics and a fundamental concept in computer science. They form the logical basis for everything from database query languages and data structures to the specification of programming language types.



The elements of set S are a, c, e , and g .

Figure 1.1: A set S containing four elements. The objects b, d , and f are not elements of S .

Specifying a Set

There are two primary ways to describe a set: by explicitly listing its members or by defining a property that its members must satisfy.

Listing Notation (Roster Method)

The most direct way to define a set is by listing all its elements between curly braces, $\{\}$. This is known as the **roster method**.

For example:

- The set of the first five letters of the alphabet is $A = \{a, b, c, d, e\}$.
- The set of the first three positive integers is $C = \{1, 2, 3\}$.
- A set can contain different types of elements: $D = \{\text{Alice}, 42, \pi\}$.

When using the roster method, there are two fundamental rules:

1. **Order does not matter.** A set is defined only by the elements it contains, not by the sequence in which they are listed. For example, $\{1, 2, 3\}$ is the exact same set as $\{3, 1, 2\}$.
2. **Each element must be unique.** An element is either in a set or it is not. Listing an element more than once is redundant and does not change the set. For instance, the set $\{a, a, b, c, c\}$ is simply $\{a, b, c\}$.

Set-Builder Notation

When listing every element is impractical or impossible (for example, with infinite sets), we use **set-builder notation**. This method defines a set by stating a property or rule that its elements must satisfy. The notation uses a vertical bar $|$ or a colon $:$, which is read as "such that."

The general structure is:

$$\{\text{variable} \mid \text{a property the variable must satisfy}\}$$

For example:

- $A = \{l \mid l \text{ is a vowel in the English alphabet}\}$
This is read as: " A is the set of all elements l such that l is a vowel." This is another way of writing $A = \{a, e, i, o, u\}$.
- $C = \{n \mid n \in \mathbb{Z} \text{ and } 0 < n < 4\}$
This is read as: " C is the set of all numbers n such that n is an integer and n is greater than 0 and less than 4." This defines the set $\{1, 2, 3\}$.
- $E = \{x \mid x \text{ is an even integer}\}$
This defines the infinite set of all even integers: $\{\dots, -4, -2, 0, 2, 4, \dots\}$.

Set-builder notation is extremely powerful because it allows us to define large, complex, or even infinite sets with a short and precise description.

1.2 Important Sets: The Number Systems

Numbers are the foundation of mathematics and computation. The different categories of numbers we use every day, from counting items to measuring continuous values, can be formally defined as sets. Understanding these fundamental sets is crucial for any software engineer, as they underpin data types, arithmetic logic, and numerical algorithms.

The Main Number Sets

The following sets are some of the most important in mathematics and are used throughout science and engineering.

Natural Numbers (\mathbb{N}) The set of positive integers used for counting: $\{1, 2, 3, \dots\}$. Sometimes, this set is defined to include 0. Due to this ambiguity, it is often clearer to specify *positive integers* or *non-negative integers*.

Integers (\mathbb{Z}) The set of all positive and negative whole numbers, including zero: $\{\dots, -2, -1, 0, 1, 2, \dots\}$.

Rational Numbers (\mathbb{Q}) The set of all numbers that can be expressed as a fraction $\frac{p}{q}$, where p and q are integers and $q \neq 0$. This includes all integers and terminating or repeating decimals. Examples: $\frac{1}{2}$, -5 , 0.25 .

Real Numbers (\mathbb{R}) The set of all numbers on the number line. It includes both rational numbers and irrational numbers (like π or $\sqrt{2}$), which cannot be expressed as simple fractions.

Complex Numbers (\mathbb{C}) The set of all numbers that can be expressed in the form $a + bi$, where a and b are real numbers and i is the imaginary unit, satisfying $i^2 = -1$.

Visualizing the relationships between these sets is key to understanding them. A Venn diagram shows the hierarchy of how these sets are nested within one another, while a number line illustrates how they cover or populate the continuum of values.

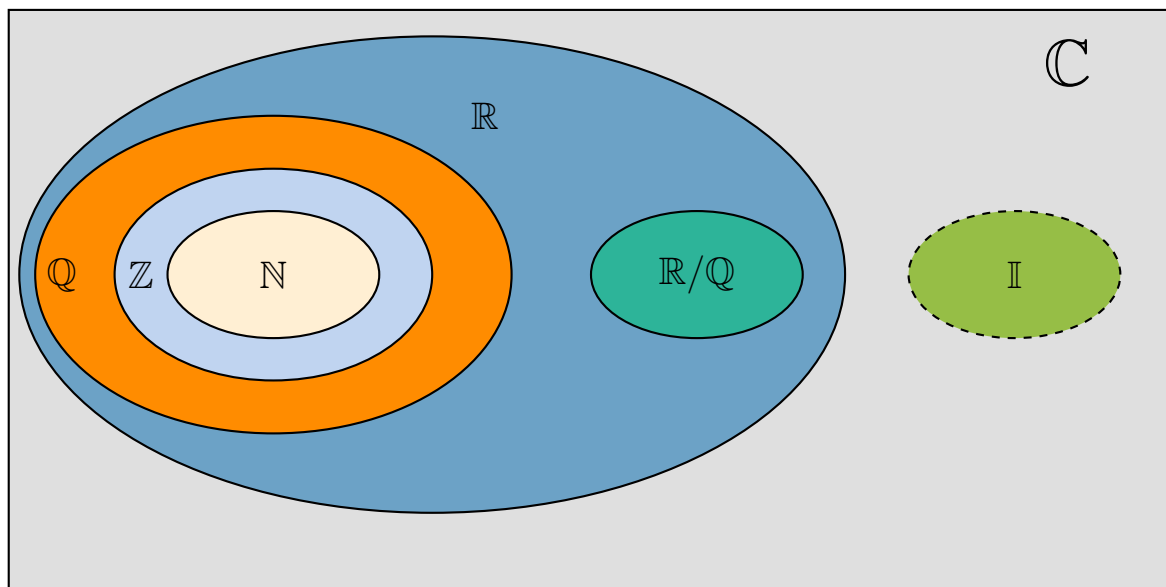


Figure 1.2: A Venn diagram illustrating the hierarchical relationship between the major number sets.

Remark: The Venn diagram in [Figure 1.2](#) shows that the set of real numbers (\mathbb{R}) is composed exclusively of rational (\mathbb{Q}) and irrational numbers. There is no real number that is not one or the other. This is why the set of irrational numbers is formally denoted as $\mathbb{R} \setminus \mathbb{Q}$, which means "the set of all real numbers, excluding the rational numbers."

Interval Notation

In many applications, we need to refer to a continuous range of real numbers. **Interval notation** is a convenient shorthand for describing such subsets of \mathbb{R} .

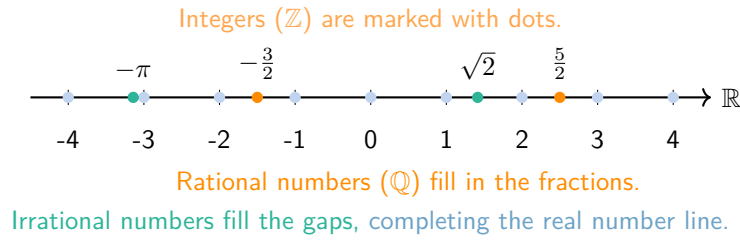


Figure 1.3: The real number line, populated by integers, rationals, and irrationals.

An interval is defined by its two endpoints. We use square brackets '[' ']' to indicate that an endpoint is included in the set, and parentheses '(' ')' to indicate that it is excluded.

Closed Interval: Includes both endpoints.

$$[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$$

Open Interval: Excludes both endpoints.

$$(a, b) = \{x \in \mathbb{R} \mid a < x < b\}$$

Half-Open Intervals: Includes one endpoint but not the other.

$$[a, b) = \{x \in \mathbb{R} \mid a \leq x < b\} \quad \text{and} \quad (a, b] = \{x \in \mathbb{R} \mid a < x \leq b\}$$

Example 1.1 Interval Notation

- $[-2, 3]$ is the set of all real numbers from -2 to 3 , including -2 and 3 .
- $(-2, 3)$ is the set of all real numbers between -2 and 3 .
- $[0, 100)$ represents all numbers from 0 up to (but not including) 100 .

Intervals can also be unbounded, extending towards positive or negative infinity (∞). Since infinity is not a number, it is always excluded with a parenthesis.

$$(a, \infty) = \{x \in \mathbb{R} \mid x > a\} \quad \text{and} \quad (-\infty, b] = \{x \in \mathbb{R} \mid x \leq b\}$$

Example 1.2 Unbounded Interval

- $(3, \infty)$ represents all real numbers greater than 3 .
- $(-\infty, 5)$ represents all real numbers less than 5 .
- $(-\infty, 0]$ represents the set of all non-positive real numbers.

1.3 Relationships Between Sets

Understanding a set is not just about its elements, but also how it relates to other sets. This section defines the fundamental relationships that allow us to compare and classify sets.

Subsets and Proper Subsets

One of the most basic relationships is that of inclusion, where one set is contained within another.

Definition 1.1 (Subset)

A set A is a **subset** of a set B if every element of A is also an element of B . We write this as $A \subseteq B$.

For example, if $A = \{1, 2\}$ and $B = \{1, 2, 3\}$, then $A \subseteq B$ because every element in A is also in B . By this definition, every set is a subset of itself (i.e., $A \subseteq A$).

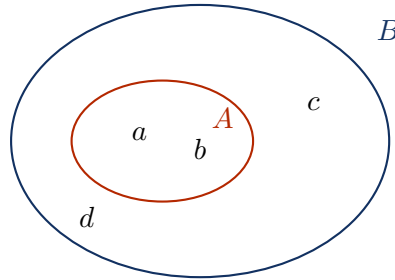


Figure 1.4: Set $A = \{a, b\}$ is a subset of set $B = \{a, b, c, d\}$, denoted $A \subseteq B$.

Sometimes we want to specify that a set is a subset of another but is not equal to it.

Definition 1.2 (Proper Subset)

A set A is a **proper subset** of a set B if $A \subseteq B$ and $A \neq B$. This means that B must contain at least one element that is not in A . We write this as $A \subset B$.

Using the previous example, since B contains the element 3 which is not in A , we can say that A is a proper subset of B , or $A \subset B$.

The Universal Set and the Empty Set

Two special sets act as the boundaries for set theory: the set containing everything and the set containing nothing.

Definition 1.3 (Universal Set)

The **universal set**, denoted by U , is the set of all possible elements under consideration in a given context. All other sets in that context are considered subsets of the universal set.

The universal set is represented in Venn diagrams by a rectangle that encloses all other sets. For example, if we are discussing integers, the universal set would be $U = \mathbb{Z}$. If we were discussing students at a university, U would be the set of all enrolled students.

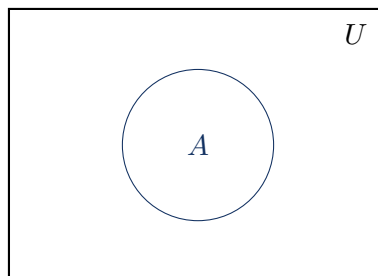


Figure 1.5: The universal set U contains all elements and sets under consideration.

Definition 1.4 (Empty Set)

The **empty set** (or **null set**) is the unique set containing no elements. It is denoted by \emptyset or by $\{\}$. ♣

The empty set has a crucial property:

The empty set is a subset of every set.

This is because there are no elements in \emptyset that are not in any other set A . Therefore, for any set A , it is always true that $\emptyset \subseteq A$.

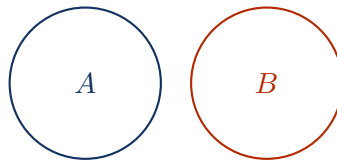
Disjoint Sets

Sometimes, sets have no relationship at all because they are entirely separate.

Definition 1.5 (Disjoint Sets)

Two sets, A and B , are **disjoint** if they have no elements in common. In other words, their intersection is the empty set: $A \cap B = \emptyset$. ♣

For example, the set of even integers $E = \{\dots, -2, 0, 2, \dots\}$ and the set of odd integers $O = \{\dots, -3, -1, 1, 3, \dots\}$ are disjoint.



Disjoint Sets

Figure 1.6: Sets A and B are disjoint because they do not overlap.

1.4 Properties of Sets

Beyond the relationships between sets, we can also describe their intrinsic properties. The two most fundamental properties are a set's size (its cardinality) and the collection of all its possible subsets (its power set).

Cardinality

The most basic property of a finite set is its size.

Definition 1.6 (Cardinality)

The **cardinality** of a finite set A , denoted $|A|$, is the number of distinct elements in the set. ♣

For example:

- If $A = \{a, b, c, d\}$, then $|A| = 4$.
- If $B = \{n \in \mathbb{Z} \mid 0 < n < 5\}$, then $B = \{1, 2, 3, 4\}$ and $|B| = 4$.
- For the empty set, $|\emptyset| = 0$.

The concept of cardinality can be extended to infinite sets, but that is a more advanced topic beyond the scope of this chapter. For our purposes, cardinality is a simple count of the elements.

Power Sets

One of the most powerful constructs in set theory is the idea of creating a set that contains all possible subsets of another set.

Definition 1.7 (Power Set)

The **power set** of a set A , denoted $\mathcal{P}(A)$, is the set of all subsets of A . The elements of the power set are themselves sets.



The power set always includes the empty set (\emptyset) and the set A itself.

Example 1.3 Finding the Power Set of $A = \{1, 2, 3\}$

To find $\mathcal{P}(A)$, we list all possible subsets of A , grouped by their cardinality:

- Subsets of size 0: $\{\emptyset\}$
- Subsets of size 1: $\{\{1\}, \{2\}, \{3\}\}$
- Subsets of size 2: $\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$
- Subsets of size 3: $\{\{1, 2, 3\}\}$

Combining all these subsets into a single set gives us the power set:

$$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

Remark:[Cardinality of a Power Set] For a finite set A with cardinality $|A| = n$, the cardinality of its power set is $|\mathcal{P}(A)| = 2^n$.

This is a crucial formula for computer science. The reason is that for each of the n elements in set A , we can make a binary choice: either we **include** it in a subset or we **exclude** it. With two choices for each of the n elements, there are $2 \times 2 \times \cdots \times 2$ (n times), or 2^n , total possible combinations, which corresponds to the total number of possible subsets.

1.5 Operations on Sets

Just as we can perform arithmetic operations on numbers, we can perform operations on sets to create new sets. These operations form the foundation of set algebra. For a software engineer, the most powerful insight is that set operations are a direct parallel to the logical operations of **Boolean algebra**. Every rule you learn for sets has an equivalent rule in logic and digital circuit design.

The Duality of Sets and Boolean Algebra

The relationship between set theory and Boolean algebra is so direct that they are considered "dually isomorphic." This means they are structurally identical. Understanding one is understanding the other. The key translations are:

Set Theory		Boolean Algebra / Logic
Union (\cup)	\iff	Boolean Sum (+) or OR
Intersection (\cap)	\iff	Boolean Product (\cdot) or AND
Complement (A^C)	\iff	Complementation (\overline{x}) or NOT
The Universal Set (U)	\iff	True (1)
The Empty Set (\emptyset)	\iff	False (0)

We will now explore the primary set operations, keeping this duality in mind.

Intersection

The intersection of two sets contains only the elements that are common to both sets. It corresponds to the logical AND operator.

Definition 1.8 (Intersection)

The **intersection** of sets A and B , denoted $A \cap B$, is the set containing all elements that are in both A and B .

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

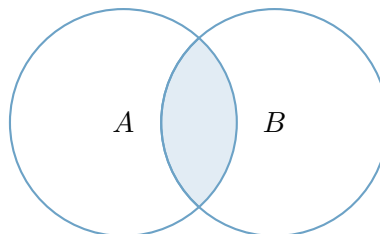


Figure 1.7: The shaded region represents the intersection $A \cap B$.

Union

The union of two sets contains all the elements that appear in either set (or both). It corresponds to the logical OR operator.

Definition 1.9 (Union)

The **union** of sets A and B , denoted $A \cup B$, is the set containing all elements that are in A , or in B , or in both.

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$



Set Difference

The difference between two sets contains the elements that are in the first set but *not* in the second set.

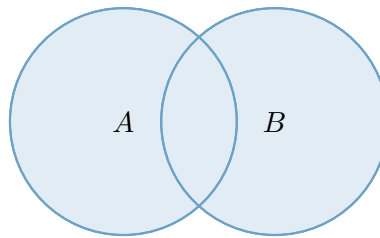


Figure 1.8: The shaded region represents the union $A \cup B$.

Definition 1.10 (Set Difference)

The **difference** of set A and set B , denoted $A \setminus B$, is the set containing all elements that are in A but not in B .

$$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$$

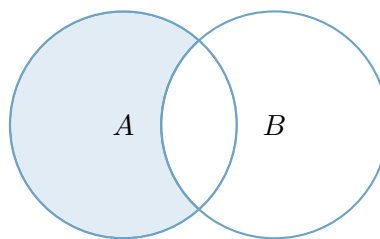


Figure 1.9: The shaded region represents the difference $A \setminus B$.

Symmetric Difference

The symmetric difference contains all elements that are in one set or the other, but not in both. It corresponds to the logical XOR operator.

Definition 1.11 (Symmetric Difference)

The **symmetric difference** of sets A and B , denoted $A \oplus B$, is the set of elements which are in either of the sets, but not in their intersection.

$$A \oplus B = (A \cup B) \setminus (A \cap B)$$

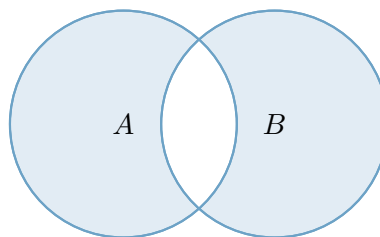


Figure 1.10: The shaded region represents the symmetric difference $A \oplus B$.

Complement

The complement of a set contains all the elements in the universal set that are *not* in the set itself. It corresponds to the logical NOT operator.

Definition 1.12 (Complement)

The **complement** of a set A , denoted A^C , is the set of all elements in the universal set U that are not in A .

$$A^C = U \setminus A$$

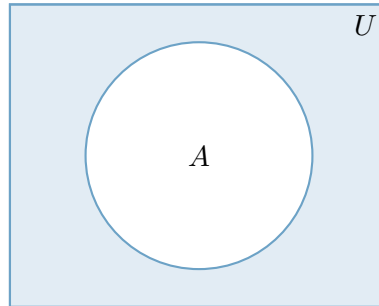


Figure 1.11: The shaded region represents the complement A^C .

1.6 Cartesian Products and Tuples

While sets are unordered collections, we often need to work with ordered collections in computer science, such as coordinates, database records, or structured data. The Cartesian product is the set operation that allows us to create these ordered structures.

Cartesian Product

The Cartesian product creates a new set from two or more existing sets, consisting of all possible ordered combinations of their elements.

Definition 1.13 (Cartesian Product)

The **Cartesian product** of sets A and B , denoted $A \times B$, is the set of all possible ordered pairs (a, b) , where the first element a is from A and the second element b is from B .

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$

The name comes from the Cartesian coordinate system, where any point on a 2D plane can be represented by an ordered pair (x, y) from the Cartesian product of the real numbers, $\mathbb{R} \times \mathbb{R}$.

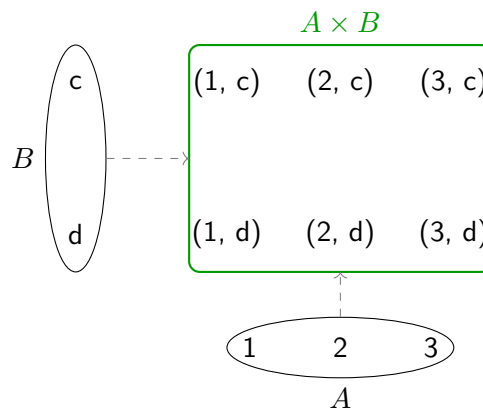


Figure 1.12: The Cartesian product of $A = \{1, 2, 3\}$ and $B = \{c, d\}$ results in a set of 6 ordered pairs.

Remark: The order of the sets in a Cartesian product matters. The set $A \times B$ is generally not equal to the set $B \times A$. For example, the pair $(1, c)$ is in $A \times B$, but the pair $(c, 1)$ would be in $B \times A$.

Tuples

The elements of a Cartesian product are called **tuples**. If the product involves n sets, its elements are called **n-tuples**.

- A **2-tuple**, such as (a, b) , is more commonly known as an **ordered pair**.
- A **3-tuple** has the form (a, b, c) .
- An **n-tuple** has the form (a_1, a_2, \dots, a_n) .

The defining characteristic of a tuple is that **order matters**. This makes tuples fundamentally different from sets.

- For sets: $\{1, 2, 3\} = \{3, 2, 1\}$
- For tuples: $(1, 2, 3) \neq (3, 2, 1)$

This property makes tuples ideal for representing data where the position of an element carries meaning, such as a database record '(UserID, Name, Email)'.

1.7 Proving Set Equalities

In software development, simplifying complex conditional logic is crucial for writing efficient and readable code. Similarly, in set theory, we often need to prove that two different expressions describe the exact same set. There are two primary methods for this: using membership tables and applying set identities.

Method 1: Membership Tables

A membership table is a tool used to prove that two set expressions are equal by checking every possible combination of an element's membership in the constituent sets.

This method is the set-theory equivalent of using a **truth table** in Boolean algebra. Instead of checking for TRUE or FALSE, we check if an element is a member (represented by a 1) or not a member (represented by a 0) of a set. If the columns for two different set expressions are identical in all rows, the expressions are proven to be equal.

Example 1.4 Showing that $A \cap B = B \setminus (B \setminus A)$

To prove this equality, we construct a membership table for all combinations of membership in sets A and B .

A	B	$A \cap B$	$B \setminus A$	$B \setminus (B \setminus A)$
1	1	1	0	1
1	0	0	0	0
0	1	0	1	0
0	0	0	0	0

Table 1.1: Fundamental Set Identities and their Boolean Algebra Counterparts.

Identity Name	Set Identity	Boolean Identity
Identity Laws	$A \cup \emptyset = A$ $A \cap U = A$	$x + 0 = x$ $x \cdot 1 = x$
Domination Laws	$A \cup U = U$ $A \cap \emptyset = \emptyset$	$x + 1 = 1$ $x \cdot 0 = 0$
Idempotent Laws	$A \cup A = A$ $A \cap A = A$	$x + x = x$ $x \cdot x = x$
Complement Laws	$A \cup A^C = U$ $A \cap A^C = \emptyset$	$x + \bar{x} = 1$ $x \cdot \bar{x} = 0$
Commutative Laws	$A \cup B = B \cup A$ $A \cap B = B \cap A$	$x + y = y + x$ $x \cdot y = y \cdot x$
Associative Laws	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$	$(x + y) + z = x + (y + z)$ $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
Distributive Laws	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ $x + (y \cdot z) = (x + y) \cdot (x + z)$
De Morgan's Laws	$(A \cap B)^C = A^C \cup B^C$ $(A \cup B)^C = A^C \cap B^C$	$\overline{x \cdot y} = \bar{x} + \bar{y}$ $\overline{x + y} = \bar{x} \cdot \bar{y}$
Absorption Laws	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	$x + (x \cdot y) = x$ $x \cdot (x + y) = x$

Conclusion: Since the column for $A \cap B$ is identical to the column for $B \setminus (B \setminus A)$ for all possible membership combinations, the two expressions are equal.

Method 2: Set Identities

While membership tables are effective, they can become very large as the number of sets increases. A more algebraic approach is to use **set identities** to simplify one expression until it matches the other. These identities are fundamental laws that govern how set operations behave.

The table below shows the most important set identities. Notice the striking similarity to the laws of Boolean algebra—they are structurally identical.

Example 1.5 Proving $A \cup (B \cap A^C) = A \cup B$ using identities

$$\begin{aligned}
 A \cup (B \cap A^C) &= (A \cup B) \cap (A \cup A^C) \\
 &= (A \cup B) \cap U \\
 &= A \cup B
 \end{aligned}$$

by Distributive Law

by Complement Law

by Identity Law

1.8 Computer Representation of Sets

While set theory provides the abstract language for collections, computer science requires concrete and efficient ways to implement these ideas. For finite universal sets, one of the most elegant and performant methods is to represent sets using **bit strings** (also known as bitmasks or bit vectors).

This technique requires two conditions:

1. The universal set U must be finite.
2. The elements of U must have a fixed, agreed-upon order.

Let $U = \{a_1, a_2, \dots, a_n\}$. Any subset $A \subseteq U$ can be represented by a bit string of length n , where the i -th bit is 1 if $a_i \in A$, and 0 if $a_i \notin A$.

Example 1.6 Representing Sets as Bit Strings

Let the universal set be $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

- The set $A = \{1, 3, 4, 8\}$ is represented by the bit string 10110001.
- The set $B = \{2, 3, 8\}$ is represented by the bit string 01100001.
- The set of all even numbers, $\{2, 4, 6, 8\}$, is 01010101.
- The empty set, \emptyset , is 00000000.

The true power of this representation is that set operations map directly to extremely fast, low-level bitwise operations that processors can execute in a single cycle.

Set Operations as Bitwise Operations

Let the bit strings for sets A and B be s_A and s_B .

- **Union** ($A \cup B$) corresponds to a bitwise OR operation.
- **Intersection** ($A \cap B$) corresponds to a bitwise AND operation.
- **Complement** (A^C) corresponds to a bitwise NOT (one's complement) operation.
- **Symmetric Difference** ($A \oplus B$) corresponds to a bitwise XOR operation.

Example 1.7 Performing Set Operations with Bit Strings

Using $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$, let $A = \{1, 3, 4, 8\}$ and $B = \{2, 3, 8\}$.

	Set Representation	Bit String Representation
Set A	$\{1, 3, 4, 8\}$	10110001
Set B	$\{2, 3, 8\}$	01100001
$A \cup B$	$\{1, 2, 3, 4, 8\}$	10110001 OR 01100001 = 11110001
$A \cap B$	$\{3, 8\}$	10110001 AND 01100001 = 00100001
A^C	$\{2, 5, 6, 7\}$	NOT 10110001 = 01001110

This bit string representation is fundamental in many areas of computing, including file permissions in operating systems, database indexing, network protocols, and graphics programming, as it provides a way

to manage and query collections with maximum efficiency.

Chapter 2 Combinatorics and Probability Theory

Imagine you are tasked with forming teams of 3 for a semester project in a class of 45 students. Initially, the order in which you choose the team members does not matter, so you are just concerned with combinations. The number of ways to form a team of 3 from 45 students comes out to 14,190 possibilities!

The following semester introduces the students to Scrum project management, where each team must have three specific roles: Scrum Master, Product Owner, and Development Team. This small change, specifying roles, suddenly transforms the problem from a simple *combination* into a *permutation*. Now, the number of possible ways to assign these roles leaps to 85,140!

Frustrated by the sheer number of options, the 45 students throw a party to relax. Being well-mannered, they decide that everyone should shake hands with every other person exactly once. After a few minutes, they calculate the total number of handshakes — 990. The students are once again surprised by how something as simple as shaking hands can add up so quickly.

As the night progresses, one student proposes a fun game — a random drawing for five door prizes, each unique. With 45 students in attendance and only five prizes available, the chance of winning nothing becomes a concerning 89 per cent. The students quickly realize that the odds are not in their favor.

Not ready to give up on their luck, a smaller group decides to flip a coin 10 times, with the hope of landing exactly five heads to win the game. However, when they learn that the probability of this happening is only about 25 per cent, their spirits dampen further.

The students conclude that rather than relying on chance, it is time to dive deeper into understanding combinatorics and probability theory. Armed with this knowledge, they can better predict outcomes and avoid future disappointments at both parties and project planning.

2.1 Sample Space and Events

A **random experiment** is one that can lead to different outcomes, even when repeated under the same conditions. This randomness is a fundamental aspect of many engineering tasks.

Definition 2.1 (Random Experiment)

A random experiment is one that can give different results, even if you do everything the same each time.



To model and analyze a random experiment, it is crucial to understand the set of possible outcomes that can occur. In probability theory, this set is called the **sample space**, denoted by S . A sample space can be either **discrete** (consisting of a finite or countably infinite set of outcomes) or **continuous** (containing an interval of real numbers). An **outcome** is a single possible result of the random experiment, and an **event** is any subset of the sample space.

Example 2.1 Network Latency

Consider an experiment where you measure the latency of data packets in a network. The sample space can be

defined based on the type of measurements:

- If latency is measured as a positive real number, the sample space is continuous: $S = \{x \mid x > 0\}$.
- If it is known that latency ranges between 10 and 100 milliseconds, the sample space can be refined to:
 $S = \{x \mid 10 \leq x \leq 100\}$.
- If the objective is to categorize latency as low, medium, or high, the sample space becomes discrete:
 $S = \{\text{low, medium, high}\}$.

Definition 2.2 (Sample Spaces, Outcomes, and Events)

Sample Space: The set of all possible outcomes of a random experiment, denoted by S .

Outcome: A single possible result of a random experiment.

Event: Any subset of the sample space, which may consist of one or more outcomes.



2.2 Types of Events in Probability Theory

The study of probability and set theory are deeply connected. In probability, we work with events, which are subsets of a sample space. The operations on these events correspond directly to the set and Boolean algebra operators we have already seen.

Union and Logical OR

The **union** of two events A and B , denoted $A \cup B$, represents the event that at least one of the events occurs. This corresponds to the logical OR operator.

Intersection and Logical AND

The **intersection** of two events A and B , denoted $A \cap B$, represents the event that both A and B occur simultaneously. This corresponds to the logical AND operator.

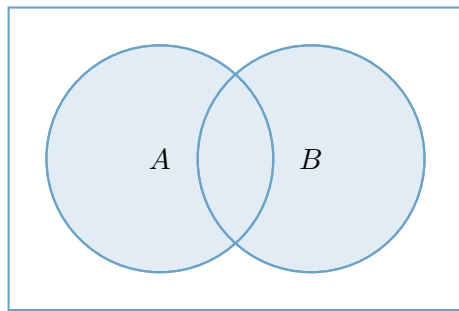
Complement and Logical NOT

The **complement** of an event A , denoted A^C , represents all outcomes in the sample space that are not in A . This operation is analogous to the logical NOT operator.

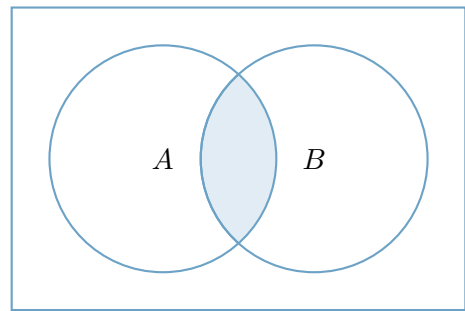
Remark: The complement of event A is most commonly written as A^C . You may also encounter the notations \bar{A} or A' in other literature. This book will consistently use A^C .

Additional Events

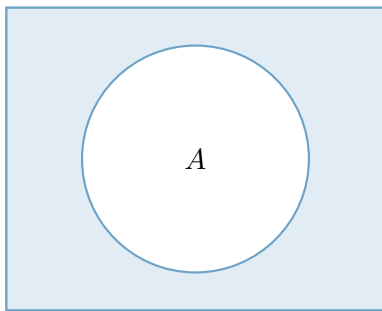
- **Difference ($A - B$):** The event that occurs if A happens but B does not.
- **Symmetric Difference ($A \triangle B$):** The event that occurs if exactly one of A or B happens, but not both. This corresponds to the logical XOR operator.
- **Mutually Exclusive Events (Disjoint):** Two events are mutually exclusive if they cannot occur at the same time ($A \cap B = \emptyset$).



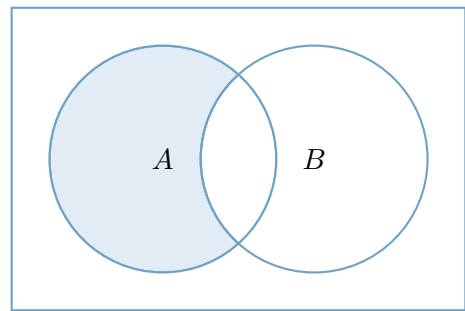
(a) Union ($A \cup B$)



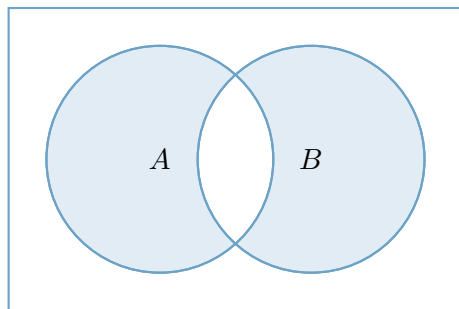
(b) Intersection ($A \cap B$)



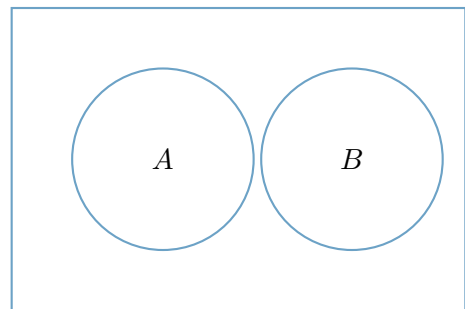
(c) Complement (A^C)



(d) Difference ($A - B$)



(e) Symmetric Difference ($A \triangle B$)



(f) Mutually Exclusive

Figure 2.1: Venn diagrams illustrating set operations.

The table below summarizes the direct correspondence between the operators used in Set Theory, Boolean Algebra, and Logic.

Operation	Set Theory	Boolean Algebra	Logic
NOT	A^C	\bar{x}	$\neg x$
OR	\cup	$+$	\vee
AND	\cap	\cdot	\wedge
NAND	$(A \cap B)^C$	$\overline{x \cdot y}$	$\neg(x \wedge y)$
NOR	$(A \cup B)^C$	$\overline{x + y}$	$\neg(x \vee y)$
XOR	$A \triangle B$	$x \oplus y$	$(x \wedge \neg y) \vee (\neg x \wedge y)$
Difference	$A - B$	$x \cdot \bar{y}$	$x \wedge \neg y$

Table 2.1: Comparison of Operators in Set Theory, Boolean Algebra, and Logic

2.3 Counting Principles

In many problems, determining the number of ways certain events can occur is essential. Counting techniques, such as permutations and combinations, help us quantify these possibilities systematically.

Multiplication Rule

The most basic counting principle is the **multiplication rule**.

Theorem 2.1 (Multiplication Rule)

If an operation consists of a sequence of k steps, and there are n_1 ways to do the first step, n_2 ways to do the second step, and so on, then the total number of ways to complete the operation is $n_1 \times n_2 \times \cdots \times n_k$.



Permutations (Order Matters)

A permutation is an arrangement of objects in a specific order.

Definition 2.3 (Permutations of Subsets)

The number of permutations of r elements selected from a set of n elements is

$$P_r^n = \frac{n!}{(n-r)!}$$



Example 2.2 There are 10 entries in a contest. The 1st, 2nd, and 3rd place prizes are awarded. How many possible results are there?

Solution: The order of selection matters, so we use permutations. The number of ways to award the prizes is the number of permutations of 3 objects selected from 10:

$$P_3^{10} = \frac{10!}{(10-3)!} = \frac{10!}{7!} = 10 \times 9 \times 8 = 720$$

Combinations (Order Does Not Matter)

A combination is a selection of objects where the order does not matter.

Definition 2.4 (Combinations)

The number of combinations of r elements selected from a set of n elements is given by

$$C_r^n = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

The formula is derived by taking the number of permutations (P_r^n) and dividing by $r!$, which is the number of ways to arrange the r selected items. This division effectively removes the duplicates created by ordering.

Example 2.3 A circuit board has four locations. If three identical components are to be placed on the board, how many different designs are possible?

Solution: Since the components are identical, the order of placement does not matter. We need to choose 3 locations out of 4.

$$\binom{4}{3} = \frac{4!}{3!(4-3)!} = \frac{4!}{3! \cdot 1!} = \frac{4}{1} = 4$$

There are 4 possible designs.

Example 2.4 Maria has three tickets for a concert and wants to invite two of her four friends (Ann, Beth, Chris, Dave). How many ways can she choose 2 friends?

Solution: The order in which she chooses her friends does not matter, so we use combinations.

$$\binom{4}{2} = \frac{4!}{2!(4-2)!} = \frac{4 \cdot 3}{2 \cdot 1} = 6$$

There are 6 possible pairs of friends she can invite.

2.4 Probability Basics

Probability quantifies the likelihood of an outcome. When all outcomes in a finite sample space are equally likely, we can use a straightforward formula to calculate the probability of an event.


Definition 2.5 (Probability with Equally Likely Outcomes)

For a random experiment where all outcomes in the finite sample space S are equally likely, the probability of an event A is the ratio of the number of favorable outcomes to the total number of outcomes.

$$P(A) = \frac{\text{Number of outcomes in } A}{\text{Total number of outcomes in } S} = \frac{|A|}{|S|}$$


All probability assignments, regardless of the scenario, must adhere to three fundamental rules known as the Axioms of Probability.

Axiom 2.1 (Axioms of Probability)

- **Axiom 1:** For any event A , $0 \leq P(A) \leq 1$.
- **Axiom 2:** $P(S) = 1$.
- **Axiom 3:** If A_1, A_2, \dots are disjoint events, then $P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots$ 

From the axioms, we can derive several useful rules for calculating probabilities of joint events, which are formed by applying set operations to individual events.

Theorem 2.2 (Rules of Probability)

- **Complement Rule:** $P(A^C) = 1 - P(A)$
 - **Addition Rule:** $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
 - **Difference Rule:** $P(A - B) = P(A) - P(A \cap B)$
 - **Subset Rule:** If $A \subseteq B$, then $P(A) \leq P(B)$.
 - **Empty Set Rule:** $P(\emptyset) = 0$.
- 

We conclude this section with a few examples illustrating the application of these rules.

Example 2.5 A company has bid on two large construction projects. Let A be the event of winning the first contract and B be the event of winning the second. The company president believes that $P(A) = 0.6$, $P(B) = 0.4$, and the probability of winning both is $P(A \cap B) = 0.2$.

- What is the probability that the company wins at least one contract?
- What is the probability that the company wins the first but not the second?
- What is the probability that the company wins neither contract?
- What is the probability that the company wins exactly one contract?

Solution:

- The probability of winning at least one contract is $P(A \cup B)$. Using the Addition Rule:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.6 + 0.4 - 0.2 = 0.8$$


- The probability of winning the first but not the second is $P(A - B)$. Using the Difference Rule:

$$P(A - B) = P(A) - P(A \cap B) = 0.6 - 0.2 = 0.4$$

- The probability of winning neither is the complement of winning at least one, $P((A \cup B)^C)$. Using the Complement Rule:

$$P((A \cup B)^C) = 1 - P(A \cup B) = 1 - 0.8 = 0.2$$

- Winning exactly one contract is the symmetric difference, $P(A \triangle B)$, which can be calculated as $P(A \cup B) - P(A \cap B)$:

$$P(A \triangle B) = 0.8 - 0.2 = 0.6$$


Example 2.6 Let A be the event that it rains today and B be the event that it rains tomorrow. We are given:

- $P(A) = 0.6$ (60% chance of rain today).
- $P(B) = 0.5$ (50% chance of rain tomorrow).

- There is a 30% chance it does not rain either day. This means the probability of no rain today AND no rain tomorrow is $P(A^C \cap B^C) = 0.3$.

Calculate the following probabilities:

- It will rain today or tomorrow.
- It will rain today and tomorrow.
- It will rain today but not tomorrow.
- It will rain on exactly one of the two days.

Solution:

- The probability of rain today or tomorrow is $P(A \cup B)$. We can find this using the given information about no rain. By De Morgan's Law, the event "no rain on either day" ($A^C \cap B^C$) is the complement of "rain on at least one day" ($(A \cup B)^C$).

$$\begin{aligned} P(A \cup B) &= 1 - P((A \cup B)^C) && \text{by Complement Rule} \\ &= 1 - P(A^C \cap B^C) && \text{by De Morgan's Law} \\ &= 1 - 0.3 = 0.7 \end{aligned}$$

The probability it rains on at least one of the days is 0.7.

- The probability of rain today and tomorrow is $P(A \cap B)$. We use the Addition Rule, rearranged to solve for the intersection:

$$\begin{aligned} P(A \cap B) &= P(A) + P(B) - P(A \cup B) \\ &= 0.6 + 0.5 - 0.7 = 0.4 \end{aligned}$$

The probability it rains on both days is 0.4.

- The probability of rain today but not tomorrow is $P(A - B)$. Using the Difference Rule:

$$\begin{aligned} P(A - B) &= P(A) - P(A \cap B) \\ &= 0.6 - 0.4 = 0.2 \end{aligned}$$

The probability it rains today but not tomorrow is 0.2.

- The probability of rain on exactly one day is the symmetric difference, $P(A \triangle B)$. This is the probability of (rain today and not tomorrow) OR (rain tomorrow and not today). Since these two events are disjoint, we can add their probabilities: $P(A - B) + P(B - A)$.

First, we find $P(B - A)$:

$$P(B - A) = P(B) - P(A \cap B) = 0.5 - 0.4 = 0.1$$

Now, add the two disjoint probabilities:

$$P(A \triangle B) = P(A - B) + P(B - A) = 0.2 + 0.1 = 0.3$$

The probability it rains on exactly one of the days is 0.3.



Bibliography

- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2022). *Introduction to algorithms* (4th). MIT Press.
- Lay, D. (2003). *Linear algebra and its applications*. Pearson Education.
- Montgomery, D. (2013). *Applied statistics and probability for engineers, 6th edition*. John Wiley; Sons, Incorporated.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Kappa Research, LLC.
- Rosen, K. H. (2012). *Discrete mathematics and its applications* (7th). McGraw-Hill Education.

Appendix A: Summation

This appendix offers methods for evaluating summations, which occur frequently in the analysis of algorithms. Many of the formulas here appear in any calculus text, but you will find it convenient to have these methods compiled in one place. The content of this appendix is based on the book *Introduction to Algorithms*, (Cormen et al., 2022). The Appendix is concluded with a useful cheat sheet for time complexity analysis.

When an algorithm contains an iterative control construct such as a **while** or **for** loop, you can express its running time as the sum of the times spent on each execution of the body of the loop. For example, in ?? we argued that the i 'th iteration of QUADRATIC-SUM took time proportional to i in the worst case. Adding up the time spent on each iteration produced the summation (or series) $\sum_{i=1}^n i$. Evaluating this summation resulted in a bound of $\mathcal{O}(n^2)$ on the worst-case running time of the algorithm. This example illustrates why you should know how to manipulate summations. Before going into details, we provide an overview of the most important summations and their closed forms.

Sum	Closed Form
$\sum_{k=0}^n ar^k \ (r \neq 1)$	$\frac{ar^{n+1} - a}{r - 1}$
$\sum_{k=1}^n k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^n k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^n k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^n x^k \ (x \neq 1)$	$\frac{x^{n+1} - 1}{x - 1}$
$\sum_{i=1,2,4,\dots,n} i$	$2n - 1$
$\sum_{k=0}^{\infty} x^k, x < 1$	$\frac{1}{1 - x}$
$\sum_{k=1}^{\infty} kx^{k-1}, x < 1$	$\frac{1}{(1 - x)^2}$
$\sum_{k=1}^n \frac{1}{k}$	$\ln n + O(1)$
$\sum_{k=1}^n (a + bk)$	$an + b\frac{n(n+1)}{2}$

Table A.1: Some Useful Summation Formulae

Summation Notation

Consider a sequence of numbers a_1, a_2, \dots, a_n , where n is a nonnegative integer. The sum of this sequence, $a_1 + a_2 + \dots + a_n$, can be represented by the notation $\sum_{k=1}^n a_k$. When $n = 0$, this summation is defined to have a value of 0. The result of a finite sum is always well-defined, and the order in which the terms are summed does not affect the final value.

For an infinite sequence a_1, a_2, \dots of numbers, we represent their infinite sum $a_1 + a_2 + \dots$ by the notation $\sum_{k=1}^{\infty} a_k$, which corresponds to $\lim_{n \rightarrow \infty} \sum_{k=1}^n a_k$. If this limit exists, the series is said to converge; otherwise, it diverges. Unlike finite sums, the terms of a convergent series cannot necessarily be rearranged without affecting the outcome. However, in an absolutely convergent series—one where $\sum_{k=1}^{\infty} |a_k|$ also converges—the terms can be reordered without changing the sum.

$$\sum_{i=i_0}^n ca_i = c \sum_{i=i_0}^n a_i$$

where c is any number. So, we can factor constants out of a summation.

Similarly, we can split a summation into two summations:

$$\sum_{i=i_0}^n (a_i \pm b_i) = \sum_{i=i_0}^n a_i \pm \sum_{i=i_0}^n b_i$$

Note that we started the series at i_0 to denote the fact that they can start at any value of i that we need them to. Also note that while we can break up sums and differences as we did above we can't do the same thing for products and quotients. In other words,

$$\sum_{i=i_0}^n (a_i b_i) \neq \left(\sum_{i=i_0}^n a_i \right) \left(\sum_{i=i_0}^n b_i \right) \quad \text{and} \quad \sum_{i=i_0}^n \frac{a_i}{b_i} \neq \frac{\sum_{i=i_0}^n a_i}{\sum_{i=i_0}^n b_i}.$$

Linearity of Summation

For any real number c and any finite sequences a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n ,

$$\sum_{k=1}^n (ca_k + b_k) = c \sum_{k=1}^n a_k + \sum_{k=1}^n b_k.$$

The linearity property also applies to infinite convergent series.

The linearity property applies to summations incorporating asymptotic notation. For example,

$$\sum_{k=1}^n \Theta(f(k)) = \Theta \left(\sum_{k=1}^n f(k) \right).$$

In this equation, the Θ -notation on the left-hand side applies to the variable k , but on the right-hand side, it applies to n . Such manipulations also apply to infinite convergent series.

Arithmetic Series

The summation

$$\sum_{k=1}^n k = 1 + 2 + \cdots + n$$

is an arithmetic series and has the value

$$\sum_{k=1}^n k = \frac{n(n+1)}{2} \quad (\text{A.1})$$

$$= \Theta(n^2). \quad (\text{A.2})$$

A general arithmetic series includes an additive constant $a \geq 0$ and a constant coefficient $b > 0$ in each term, but has the same total asymptotically:

$$\sum_{k=1}^n (a + bk) = \Theta(n^2). \quad (\text{A.3})$$

Sums of Squares and Cubes

The following formulas apply to summations of squares and cubes:

$$\sum_{k=0}^n k^2 = \frac{n(n+1)(2n+1)}{6}, \quad (\text{A.4})$$

$$\sum_{k=0}^n k^3 = \frac{n^2(n+1)^2}{4}. \quad (\text{A.5})$$

Geometric Series

For real $x \neq 1$, the summation

$$\sum_{k=0}^n x^k = 1 + x + x^2 + \cdots + x^n$$

is a geometric series and has the value

$$\sum_{k=0}^n x^k = \frac{x^{n+1} - 1}{x - 1}. \quad (\text{A.6})$$

The infinite decreasing geometric series occurs when the summation is infinite and $|x| < 1$:

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1 - x}. \quad (\text{A.7})$$

If we assume that $0^0 = 1$, these formulas apply even when $x = 0$.

A **special** case of the geometric series arises when the terms form a progression of powers of 2. That is, the terms are $1, 2, 4, 8, \dots, n$. In this case:

$$\sum_{i=1,2,4,\dots,n} i = 1 + 2 + 4 + 8 + \cdots + n.$$

This series can be derived from the general formula for a geometric series. The general form of a geometric series is:

$$S = a + ar + ar^2 + \cdots + ar^{k-1},$$

where:

- a is the first term ($a = 1$ in this case),
- r is the common ratio ($r = 2$ here),
- k is the number of terms in the series.

The number of terms, k , is determined by how many times the progression $1, 2, 4, \dots, n$ doubles until reaching n . Since $n = 2^{k-1}$, we have $k = \log_2(n) + 1$.

Using the geometric series formula (A.6):

$$\sum_{i=1,2,4,\dots,n} i = \frac{2^k - 1}{2 - 1} = 2^k - 1.$$

Substituting $k = \log_2(n) + 1$:

$$\sum_{i=1,2,4,\dots,n} i = 2^{\log_2(n)+1} - 1 = 2n - 1.$$

Thus, the sum of the powers of 2 up to n is:

$$\sum_{i=1,2,4,\dots,n} i = 2n - 1.$$

Harmonic Series

For positive integers n , the n th harmonic number is

$$\begin{aligned} H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} \\ &= \sum_{k=1}^n \frac{1}{k} \end{aligned} \tag{A.8}$$

$$= \ln n + O(1). \tag{A.9}$$

Integrating and Differentiating

Integrating or differentiating the formulas above yields additional formulas. For example, differentiating both sides of the infinite geometric series (A.7) and multiplying by x gives

$$\begin{aligned} \sum_{k=0}^{\infty} kx^k &= \frac{x}{(1-x)^2} \\ \text{for } |x| < 1. \end{aligned} \tag{A.10}$$

Telescoping Series

For any sequence a_0, a_1, \dots, a_n

$$\sum_{k=1}^n (a_k - a_{k-1}) = a_n - a_0, \quad (\text{A.11})$$

since each of the terms a_1, a_2, \dots, a_{n-1} is added in exactly once and subtracted out exactly once. We say that the sum telescopes. Similarly,

$$\sum_{k=0}^{n-1} (a_k - a_{k+1}) = a_0 - a_n.$$

As an example of a telescoping sum, consider the series

$$\sum_{k=1}^{n-1} \frac{1}{k(k+1)}$$

Rewriting each term as

$$\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1},$$

gives

$$\begin{aligned} \sum_{k=1}^{n-1} \frac{1}{k(k+1)} &= \sum_{k=1}^{n-1} \left(\frac{1}{k} - \frac{1}{k+1} \right) \\ &= 1 - \frac{1}{n}. \end{aligned} \quad (\text{A.12})$$

Reindexing Sums

A series can sometimes be simplified by changing its index, often reversing the order of summation. Consider the series $\sum_{k=0}^n a_{n-k}$. Because the terms in this summation are a_n, a_{n-1}, \dots, a_0 , we can reverse the order of indices by letting $j = n - k$ and rewrite this summation as

$$\sum_{k=0}^n a_{n-k} = \sum_{l=0}^n a_j$$

Generally, if the summation index appears in the body of the sum with a minus sign, it's worth thinking about reindexing.

As an example, consider the summation

$$\sum_{k=1}^n \frac{1}{n-k+1}. \quad (\text{A.13})$$

The index k appears with a negative sign in $\frac{1}{(n-k+1)}$. And indeed, we can simplify this summation, this time setting $j = n - k + 1$, yielding

$$\sum_{k=1}^n \frac{1}{n-k+1} = \sum_{j=1}^n \frac{1}{j}, \quad (\text{A.14})$$

which is just the harmonic series (A.8).

Products

The finite product $a_1 a_2 \dots a_n$ can be expressed as

$$\prod_{k=1}^n a_k.$$

If $n = 0$, the value of the product is defined to be 1. You can convert a formula with a product to a formula with a summation by using the identity

$$\log \left(\prod_{k=1}^n a_k \right) = \sum_{k=1}^n \log a_k.$$

Cheat Sheet for Time Complexity Analysis

This section provides essential formulas and shortcuts for asymptotic analysis and algorithm evaluation. These complement the detailed summation methods in this appendix.

- **Logarithmic Factorial Approximation (Stirling's Approximation):**

$$\log(n!) \approx n \log n - n.$$

- **Logarithmic Exponent Rule:**

$$2^{\log_2 n} = n.$$

- **Base as a Power with Logarithmic Exponent:** If the base is a^b and the exponent involves \log_a , then:

$$(a^b)^{\log_a(n^c)} = n^{bc}.$$

Example: $4^{\log_2(n^2)} = n^4$ (since $4 = 2^2$).

- **Power of a Logarithm Rule:**

$$\log(n^k) = k \log n, \quad \text{for } k > 0.$$

- **Logarithmic Growth for Powers of 2:** For $n = 2^k$,

$$k = \log_2 n.$$

- **Simplifying Logarithmic Expressions:**

$$\log_a n = \frac{\log_b n}{\log_b a}, \quad \text{for } a, b > 0.$$

- **Common Asymptotic Comparisons:**

$$n^c \ll c^n \quad \text{for any constant } c > 1.$$

- **Factorial Asymptotics (Expanded Stirling's Approximation):** Using Stirling's formula, the growth of $n!$ is approximately:

$$n! = \Theta \left(\sqrt{2\pi n} \left(\frac{n}{e} \right)^n \right).$$

- **Sum of Powers of 2:** If the terms in a series form powers of 2 (e.g., $1, 2, 4, 8, \dots, n$), the sum is:

$$\sum_{i=1,2,4,\dots,n} i = 2n - 1.$$